JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 1007-1023 (2017)
DOI: 10.6688/JISE.2017.33.4.10

# An Attribute-Based Trust Negotiation Protocol for D2D Communication in Smart City Balancing Trust and Privacy*

JINGJING GUO[1], JIANFENG MA[1,2], XINGHUA LI[1,+], JUNWEI ZHANG[1]
AND TAO ZHANG[3]
[1]*School of Cyber Engineering*
[2]*The State Key Laboratory of Integrated Services Networks*
[3]*School of Computer*
*Xidian University*
*Xi'an, Shaanxi Province, 710071 P.R. China*
*E-mail: {jjguo; jwzhang; taozhang}@xidian.edu.cn; {jfma; xhli1}@mail.xidian.edu.cn*

Smart city is an urban development vision to integrate multiple information and communication technology (ICT) solutions in a secure fashion to manage a city's assets. It includes E-home, E-office, E-health, E-traffic and so on. All of these depend on the data collecting from multifarious devices and the following data processing and analyzing. So, communication between various devices (such as smartphone and so on) will be very frequent. In such an environment, the trust relationship between devices will be an important premise to guarantee an interaction can be carried on successfully. In this paper, we propose an attribute-based trust negotiation scheme for communication between devices (D2D communication) in a smart city. In this paper, we modeled the trust negotiation process as a 0/1 knapsack problem. We adopt the secure two-party computation technique based on the homomorphic encryption to guarantee its security. The proposed protocol can make sure that a device satisfies its counterparty's access policy while disclosing minimal privacy due to the credential disclosing. The theoretical analysis shows that our protocol is complete and secure in the semi-honest environment. Furthermore, there is no disclosure of credentials before both participants can ensure a success negotiation strategy exists. Moreover, devices cannot obtain the policies of their counterparty in the negotiation process. Finally, we did some simulations to analyze the computation cost of our protocol under different credential scales and resource access thresholds.

***Keywords:*** smart city, trust negotiation, D2D communication, tradeoff between trust and privacy, attribute-based

## 1. INTRODUCTION

Nowadays, the city is growing increasingly larger, more complex and more important with ever increasing speed. The unprecedented growth rate creates an urgency to find smart ways to manage the accompanying challenges. One promising way is by using technologies to improve the efficiency of services and meet residents' needs, which will produce multiple smart scenarios, such as E-home, E-health and E-traffic. These scenarios will be accompanied by mass information exchange between mobile terminals and embedded devices as well as connected sensors. Meanwhile, the communication traffic will tend to transform from hop-by-hop to end-to-end. In this situation, the security in

Received July 15, 2016; revised August 30, 2016; accepted September 12, 2016.
Communicated by Zhe Liu.
+ Corresponding author.
* This work was supported by the grant from the National Natural Science Foundation of China (No. 61602360, No. 61372075 and No. 61602365), and the National High Technology Research and Development Program (863 Program) (No. 2015AA016007).

communication between devices will be a huge challenge. In such virtual and often anonymous interactions, traditional authentication schemes [1, 2] may be insufficient to guarantee the security of devices [3], because the involved devices may belong to different organizations, adopt different security architectures and without any interaction before. Before using or offering a resource, both parties wish to ensure the counterparty is trustworthy currently. In such cases, establishing trust relationship between devices is the most essential procedure to ensure an interaction can be carried on successfully.

Recently, lots of trust evaluation models have been proposed for various networks. They are used to evaluate an entity's trust level to support the evaluator's decision (such as in e-commerce) [4] or give others a recommendation (such as in social network) [5, 6]. Automated Trust Negotiation (ATN) [7] offers devices a promising way to improve the way to access information. It provides a way to establish trust relationship between two strangers, where interactions might happen between them without prior knowledge. Usually, the negotiation is carried out by digital credentials exchange between the participants. The credentials are digitally signed assertions generated by a credential issuer about certain attribute of the credential owner [8]. Considering the privacy and the risk of benefit loss, devices usually assign some access constraints to their own resources.

In existing ATN approaches [9], there are some problems, such as disclosing the possession of credentials or policies before ensuring there is a feasible negotiation strategy. In addition, some studies taking the privacy into account cannot guarantee the disclosed privacy is minimal.

In this work, we propose a bidirectional trust negotiation model for D2D communications in smart city. It can efficiently determine whether there is a feasible negotiation strategy between two entities before they disclose credentials to each other. That is to say, our protocol won't disclose any credential and access policy before both participants can ensure the negotiation will be successful. Meanwhile, the proposed protocol also enables entities to negotiate successfully with minimal privacy disclosed.

The rest of this paper is organized as follows. Section 2 outlines the related works, followed by the description of a general scenario we considered and related techniques in Section 3. In Section 4, we give the proposed protocol in detail. The properties of the protocol are given in terms of security and performance in Section 5. Some simulation results are provided in Section 6. Finally, Section 7 concludes the paper.

## 2. RELATED WORKS

There have been many researchers engaged in trust topic and proposed lots of approaches [10-13] for trust negotiation. Some trust negotiation related languages have also been developed to express the negotiation policy and credentials [14, 15].

Typical solutions for trust negotiation are proposed to trade privacy for trust, such as the eager strategy and parsimonious strategy [9]. In eager strategy, an entity discloses all the currently unlocked credentials to its counterparty. Its disadvantage is some irrelevant credentials may be disclosed unnecessarily. Conversely, the parsimonious strategy discloses credentials only after exchanging sufficient policy and finding a satisfied credential exchange sequence. It may disclose some policies unrelated to the exchange sequence.

Some approaches were also proposed taking privacy preservation into account. Seigneur and Jensen [16] gave an idea to achieve the trade-off between trust and privacy. It can ensure minimal trade of privacy for trust where entities use pseudonyms, while they didn't give a specific way to implement it. Seamons *et al.* [17] gave an overview about the privacy vulnerabilities during trust negotiation, such as possession or non-possession of a sensitive credential. In general, the privacy information in existing ATN works can mainly be divided into two categories – content sensitive and possession sensitive information. Content sensitive information including the attributes and the access policies, and the possession sensitive information referring to the information disclosed implicitly during the interaction. Bertino *et al.* proposed an approach that selectively disclosed the attributes in a credential to protect its privacy information [15]. A work related to ours is [18] which explores an approach to determine the credential set which satisfies a certain point based policy with least privacy value [18]. It used a point-based access control policy. That is less rigid than a Boolean expression, which is more suitable to the fuzzy identity of the involved device in smart city. While, this work assumes all entities reach a consensus on the trust point of a certain credential, which is not realistic for D2D communication in a smart city.

Except the previous mentioned traditional works, there are also many works emerging in recent years solving trust negotiation related problems. Zhang *et al.* proposed an XML-based trust negotiation between Web Services [19]. This protocol aims to eliminate the failed trust negotiation caused by the file format interoperability problem, so it adds a checking file formats process before the formal negotiation. In order to off-load the non-trivial computational and communications costs on servers, Adams *et al.* proposed a receipt-mode trust negotiation scheme [20]. They mitigate the trust negotiation process to delegated receipt-generating helper servers, so that the servers can only in charge of the service provision. In [21], authors proposed a trust negotiation protocol extending the traditional trust verification method. It can deal with the situation that the policy has a complex structure (such as a tree structure) rather than a simple set. Li *et al.* proposed a new security policy negotiation method [22], while it can only support the negotiation of the same kind of policy. In [23], the authors proposed a trust negotiation scheme which focused mainly on the automatic detection of policy cyclic dependencies and the repetitive credential request attacks. There is also some trust negotiation schemes proposed for specific scenarios [24, 25].

Most schemes mentioned above only considered the server-side policy matching, where the server's policies (access constraint) are tested against the client's credentials. Furthermore, they did not mention how to judge whether there is a feasible negotiation strategy before credential exchanging. In this paper, we consider policy matching in both communicating sides. By alternately repeating the server-side and client-side policy exam, a sequence can be decided if a feasible negotiation strategy exists. It represents the constraint of the credential's trust quantity disclosed each time. Then, we determine the final credential exchange sequence with minimal privacy disclosed.

## 3. PRELIMINARIES AND A TYPICAL SCENARIO

Our protocol is based on the secure multi-party computation (SMC) [26] and the dynamic programming method for the 0/1 knapsack problem [27]. SMC was introduced

first by Yao. It allows two entities with respective private inputs $a$ and $b$ to compute a function $f(a, b)$ by engaging in a secure protocol for public function $f$. The protocol reveals no additional information other than the result $f(a, b)$. In this paper, SMC can guarantee that the communicating devices only know that whether there is a feasible trust negotiation strategy without any additional information. The 0/1 knapsack problem is defined as follows: Given items with different integer values and weights, find the most valuable set of items that fit in a knapsack with fixed integer capacity. We formalize the optimal credential selecting problem as a 0/1 knapsack problem which can be solved by the dynamic programming algorithm [27].

Now we describe a general scenario considered throughout the paper. We assume a device $A$ wants to access a resource $S$ owned by device $B$. $B$ assigns access constraint for each resource it owns. Only if $A$ discloses sufficient information (credential) to $B$, $B$ will grant it access to the resource. Meanwhile, $A$ also defines some policies for revealing its credentials. It may request $B$ to reveal appropriate information before $A$ reveals information according to $B$'s policy. In the negotiation process, $A$ and $B$ consider their negotiation policies and credentials are private information. They wish to reveal no privacy information unless they can ensure the negotiation will be success. Even if there is a successful trust negotiation strategy, they want to reveal minimal private information.

Furthermore, because the devices may belong to different organizations, they may have different standpoint to the trust weight of a certain credential or information. We should guarantee the negotiation can be success in this case. To our knowledge, this is not mentioned in the existing works.

## 4. THE ATTRIBUTE-BASED TRUST NEGOTIATION PROTOCOL

In this section, we first give the overall description of the proposed protocol, and then we describe each component of the protocol in detail.

### 4.1 Bird's Eye View of the Negotiation System

We assume all the devices agree on a set of credential as the universe of credentials $C = (c1, c2, \ldots, cn)$.

The devices adopt a point-based trust management policy like [18]. Specifically, each device associates an access threshold point ($T$) with each resource (*e.g.* a service or a credential). It requires the resource requester disclose at least $T$ points trust (by disclosing credentials) to access the corresponding resource. It also defines two scores for each credential: trust score ($a$) and privacy score ($p$). The trust score reveals the utility of the information including in the credential to the negotiation and the privacy score represents the inverse of the willingness to disclose a credential.

Each device has a set of credentials which is the subset of $C$ and assigns the three parameters ($T_{ci}$, $a_{ci}$, $p_{ci}$) to credential $ci$ if the device has this credential. We assume there is a positive correlation between $T_{ci}$ and $a_{ci}$, that is to say, if $T_{ci} > T_{cj} (i \neq j)$, then $a_{ci} > a_{cj}$, and vice versa. This is because, in most cases, the more trust score a credential has, the more information it consists, and less willing it has to reveal it.

In this work, we assume each device $i$ has a public attribute set $Attr_i$ which can be

seen by anyone. Before two devices start the negotiation, both of them need to ensure they have a large enough intersection of their public attribute sets, so that they have similar viewpoint of the trust score of a certain type of credential. Each device assigns a threshold ($w$) of the attribute intersection's scale for beginning the negotiation.

The negotiation model consists of three stages:

**Attribute matching** – In this phase, two participants will check whether they have sufficient common public attribute, so that they can reach a consensus on the following negotiation process.

**Policy negotiation** – In this phase, the requesting device and the resource providing device repeat the client-side and server-side policy matching alternately. It will terminate until they find a feasible negotiation strategy or ensure that there is no successful negotiation between them. If there is a feasible strategy, they will begin the next phase.

**Credential determination** – This phase will compute an optimal credential disclosing sequence, so that minimal amount of sensitive information of both the participants will be disclosed. Their policies, meanwhile, can also be satisfied. Then, the resource requester and owner will exchange the credentials according to the calculated strategy to complete the negotiation.

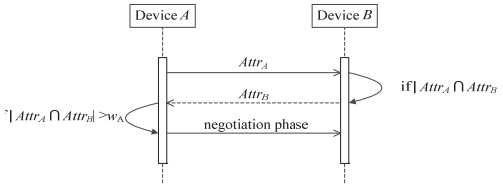
### 4.2 Attribute Matching

We assume each device maintains a public attribute set made up of several tags. Each tag indicates the attribute of the device in a certain aspect, *e.g.* the attribute set of an undergraduate student's smartphone may include gender, major, hobby and so on.

In order to get the similar attribute between a device and its counterparty, there should be an authority organization to define a universal set of the attributes. Each device will publish its public attributes based on the universal attribute set. These attributes can be seen by any device.

If two devices want to carry out a negotiation, they must ensure they have similar attitude to the utility of each credential, otherwise, the negotiation will probably fail. To this end, they should guarantee there is a large enough intersection of their public attribute sets. For example, a young male college student may think a credential including the counterparty's DotA (an online game) account can increase its trust level, while a retired female factory worker might think it is useless at all. This phenomenon is caused by the great difference between the two individuals' attributes.

The attribute matching process can be done as follows.




If device $A$ and $B$ want to carry out a negotiation, they should first measure the cardinality of their attribute sets' intersection (denoted as $|Attr_A \cap Attr_B|$). It can reflect the

similarity of their attributes. As mentioned above, device $A$ and $B$ define threshold ($w_A$ and $w_B$) beforehand to indicate the least amount of the same attribute their counterparty should has. If $|Attr_A \cap Attr_B| > \max\{w_A, w_B\}$, they will move to the policy negotiation phase. Else, it means there are too many differences between their attributes reflecting their attitude to the utility of different information. They will terminate the negotiation.

### 4.3 Policy Negotiation Protocol

In this phase, the requesting device and the resource providing device repeat the client-side and server-side policy matching alternatively. The aim is to determine the response of the client (server) $A$ according to its counterparty's ($B$) policy. There are two types of response. The first one is fail when $A$ finds there cannot be a successful negotiation between $A$ and $B$. The second one is a number ($i$). It represents $B$ must reveal a credential set which has at least $T_{ci}$ trust scores before $A$ reveals credentials to satisfy $B$'s policy ($T_{ci}$ is the access threshold of $A$'s credential $ci$).

We define the policy owned by a device as a tuple $<r, T_r>$. It means any device who wants to access resource $r$ must reveal credentials with at least $T_r$ trust score.

Algorithm 1 is the policy matching algorithm to complete the trust negotiation process. We assume a requesting device $A$ wants to access a resource $r$ owned by a providing device $B$. The output of the algorithm is $A$'s response to the $B$'s policy about $r$.

**Algorithm1:** Policy matching method in server (client)-side
**Input:** $B$'s policy $<r, T_r>$ for resource $r$; $A$'s credential set $<c1, ..., cn>$ ordered by increasing $g$ trust score.
**Output:** $A$'s response.

```
1. i = 1;
2.     sum = 0;
3.     while (i ≤ n) {
4.         sum+ = a_ci;
5.         if (sum ≥ T_r)
6.             return i;
7.         else i++;}
8. return fail;
```

In Algorithm 1, the comparison between $sum$ and $T_r$ uses the secure two-party computation protocol given in [28], so that both $a_{ci}$ and $T_r$ can keep private. Both $A$ and $B$ learn nothing in addition to the result of the policy matching. The detail of the secure two-party computation to compare two numbers can be seen in [26].

Using Algorithm 1, a device can decide the response to its counterparty. The whole negotiation phase is made up of several such policy matching processes in both requesting device and providing device side alternatively.

Now, we give the negotiation algorithm based on the policy matching method. It shows how two devices find a feasible strategy according their policies and credentials.

We assume the negotiation is between client device $A$ and resource owner device $B$ for accessing resource $R$. Both $A$ and $B$ maintain a negotiation result sequence $<n_{Ai}>$ and $<n_{Bi}>$. $n_{Ai}(n_{Bi})$ represents the trust score of the credential which is the result of the $i$th server-side (client-side) policy matching separately if the result is not fail. The negotia-

tion process is shown below.

(1) First, $A$ sends request to $B$ for accessing resource $R$.
(2) $B$ puts $T_R$ into its negotiation result sequence which indicates $A$ must disclose a credential set with at least $T_R$ trust score to $B$. Then, $B$ sends a message to $A$ that indicates the start of the server-side policy matching.
(3) $A$ determines the response to $B$'s access policy for $R$ using Algorithm 1:
   (a) If the response is fail, the negotiation is terminated;
   (b) If the response is a number representing a credential ($ci$) of $A$, add $T_{ci}$ to sequence $<n_{Ai}>$. If $T_{ci}$ is zero which indicates $A$ can immediately reveal credentials to satisfy $B$'s policy, $A$ and $B$ move to the credential determination and exchange stage;
   (c) If $T_{ci}$ isn't zero and it is the smallest item in the sequence, $A$ sends $T_{ci}$ to $B$. Then turn to step (4) to run the client-side policy matching. Else the negotiation is terminated.
(4) $B$ runs the client-side policy matching using Algorithm 1, the input parameters are $T_{ci}$ and $B$'s ordered credential set with increasing trust score.
   (a) If the result is fail, the negotiation will be terminated;
   (b) If the response is a number representing a credential ($ci$) of $B$, add $T_{ci}$ to sequence $<n_{Bi}>$. If $T_{ci}$ is zero, a feasible negotiation strategy is found, then both $A$ and $B$ move to the next stage. Else, they back to the step 3 to perform server-side policy matching with $T_{ci}$ as one of its parameters if it is the smallest item in $<n_{Bi}>$.

According to the negotiation process mentioned above, we have Lemma 1.

**Lemma 1:** If a device has two credentials $C_1$ and $C_2$ and $T_{C_1} > T_{C_2}$, $C_2$ must be earlier available than $C_1$, that means if $C_2$ is not available now, $C_1$ must not be available now.

***Proof*:** We will prove its inverse and negative proposition is true. We assume $C_1$ is available now, so the entity's counterparty must have been disclosed some credentials which have at least $T_{C_1}$ trust score. Because $T_{C_1} > T_{C_2}$, $C_2$ must be available now.

### 4.4 Credential Determination

After policy negotiation process, the two devices can ensure there must be a feasible negotiation strategy between them or not. Now both parties have a decreasing ordered negotiation result sequence. In this stage, they will determine the final credentials to exchange gradually which can satisfy their policies with minimal amount of privacy disclosed. The protocol is shown below. It begins from the party who terminates the policy negotiation. We assume $A$ terminates the policy negotiation.

(1) $A$ maintains a binary vector $<x_1, \ldots, x_n>$ as the unknown variable to be computed, where $x_i$ equals 1 if credential $ci$ will be disclosed in this round, and 0 otherwise.
(2) $A$ determines the credential set revealing to $B$ whose trust score should be at least $L_B$ (the last item in $B$'s negotiation result sequence $<n_{Bi}>$). Meanwhile, $A$ wishes disclose minimal privacy information. So, the question can be formalized as the following equation where $n$ is the amount of $A$'s credential.

$$\begin{aligned} &\min \sum_{i=1}^{n} x_i p_{ci} \\ &subject\ to\ \sum_{i=1}^{n} x_i a_{ci} \geq L_B. \end{aligned} \tag{1}$$

Eq. (1) can be rewritten to Eq. (2) by using a new variable $y_i = 1 - x_i$:

$$\begin{aligned} &\max \sum_{i=1}^{n} y_i p_{ci} \\ &subject\ to\ \sum_{i=1}^{n} y_i a_{ci} \leq \sum_{i=1}^{n} a_{ci} - L_B. \end{aligned} \tag{2}$$

We denote $\sum_{i=1}^{n} a_{ci} - L_B$ as $T'$ (marginal threshold), then Eq. (2) can be seen as a 0/1 knapsack problem and solved by dynamic programming.

(3) In this work, $A$ and $B$ consider $a_{ci}$ and $L_B$ as private information separately. Both of them won't disclose their own privacy to others, so $A$ adopts the Fingerprint protocol proposed in [18] to select the optimal credential combination.

The concrete procedure of the Fingerprint protocol is as follows:

$A$ and $B$ use the normal dynamic programming to solve the 0/1 knapsack problem shown in step 2. One different place is that they compute each entry of the dynamic programming matrix ($M_{ij}$) using homomorphic encryption [29]. Moreover, during this process, each credential's ($ci$) privacy score $p_{ci}$ is converted into another score $P_{ci}$($P_{ci}$=$p_{ci}$*$2^n$ +$2^{i-1}$). $n$ is the number of $A$'s credentials. The aim of the conversion is to solve the trace back problem in integer linear programming problems. By adopting this conversion, $A$ can trace the optimal solution from the final computed value securely and privately. More details about the protocol are described in [18]. The result of this step is that $A$ learns the optimal selection of credentials to satisfy Eq. (2), because the $i$th least significant bit of $M_{nT'}$'s binary expression is the value of $x_i$ in step 1.

(4) $A$ sends the optimal credentials computed in step (3) to $B$ and deletes these credentials from the candidate credentials in later round to avoid the duplicate selection.
(5) $B$ checks the validity of the credentials received from $A$. If they are valid, $B$ updates the penultimate item of sequence $<n_{Bi}>$ to the result of it minus the last item and deletes the last item. Then $B$ begins to compute the optimal credential set whose trust score should not be less than $L_A$ (the last item in sequence $<n_{Ai}>$). In the meanwhile, it will disclose the minimal private information. $B$ adopts the same method as $A$ described in step 3 to achieve that end. After $B$ obtaining the optimal credential set, $B$ sends the optimal credential set to $A$.
(6) $A$ checks the validity of the credentials received from $B$. If the credentials are valid, $A$ updates the penultimate item in sequence $<n_{Ai}>$ to the result of it minus the last item and deletes the last item.
(7) If both the sequence $<n_{Ai}>$ and $<n_{Bi}>$ are empty, the negotiation completes successfully, and $A$ is granted the permission to access $B$'s resource. If not, repeat the above steps (2)-(6).

# 5. PROPERTY ANALYSIS

## 5.1 Feasibility and Completeness

In this section, we analyze the feasibility and completeness of our protocol. Feasibility in our work means that the protocol can end in any case, and meanwhile the result of the protocol is reasonable. We adopt the definition of a complete negotiation protocol mentioned in [30] to measure the completeness. Authors in [30] defined that a complete negotiation protocol should be able to find a successful negotiation strategy whenever it exists. It can be proved that our protocol has the following properties.

**Lemma 2:** In the policy negotiation phase, an entity $A$ obtains a negotiation result sequence. If the later received item ($n_i$) is equal to or bigger than the former received item ($n_j$, $i > j$), there won't be a successful strategy.

***Proof*:** Assume the negotiation result sequence of $A$ is $<n_{A1}, n_{A2}, \ldots, n_{Ai}, \ldots, n_{Aj}, \ldots>$ and there has $n_{Aj} > n_{Ai}$. The policy negotiation process is shown in Fig. 1. Symbols in the parentheses represent the credential set which can satisfy the access policy (the symbols before the parentheses). Therefore, the credential disclosure sequence of $B$ is $<\ldots cre\_set_{Bj} - cre\_set_{Bj+1}, cre\_set_{Bj-1} - cre\_set_{Bj}, \ldots, cre\_set_{Bi} - cre\_set_{Bj+1}, \ldots, cre\_set_{B1} - cre\_set_{B2}>$, and the same is true for $A$.

Based on the protocol mentioned in previous section, we know: $cre\_set_{Bj}$ must be disclosed earlier than $cre\_set_{Bi}$. If $n_j > n_i$, that means $cre\_set_{Bj}$'s trust score won't be less than $cre\_set_{Bi}$'s, so $cre\_set_{Bj}$ won't be earlier available than $cre\_set_{Bi}$.

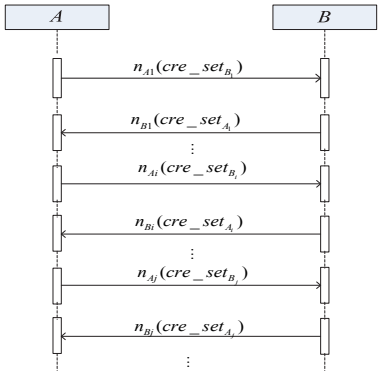



Fig. 1. Policy negotiation process between $A$ and $B$.

It is obvious there is a contradiction, so there won't be a feasible negotiation.

**Lemma 3:** The policy negotiation protocol is feasible.

***Proof*:** We consider two conditions in the policy negotiation process.

If the protocol terminates normally, the negotiation result sequences both parties received ($<n_{A1}, \ldots, n_{Ai}, \ldots>$ and $<n_{B1}, \ldots, n_{Bi}, \ldots>$) must be in descending order. That

means for each $i$, there have $n_{Ai} > n_{A(i+1)}$ and $n_{Bi} > n_{B(i+1)}$, so the items in either two sequences can be decreased to zero eventually. When an item in either of the two sequences becomes zero, a feasible negotiation strategy is found and the protocol terminates.

If either of the negotiation result sequences breaks the descending order feature, there has policy conflict and a successful trust negotiation is impossible. At this time, the protocol is also end.

**Theorem1:** Our protocol is a complete protocol.

***Proof*:** Assume there exists a feasible negotiation strategy between $A$ and $B$ and the credential disclosing begins from $A$. The final credential disclosure sequences of $A$ and $B$ are $<cre_{a_1}, \ldots, cre_{a_n}>$ and $<cre_{b_1}, \ldots, cre_{b_m}, resource>$. It is obviously that Eq. (3) is true.

$$\max_{cre \in cre_{b_x}} \{T_{cre}\} \le \sum_{i=1}^{x} \sum_{c \in cre_{a_i}} p_c \tag{3}$$

We can see credential set $\bigcup_{i=1}^{x} cre_{a_i}$ satisfies $B$'s access policy for credential set $cre_{bx}$.

In our policy negotiation protocol, the $i$th item in $A$'s negotiation result sequence is the least trust score of the credential set its counterparty should disclose to it in the first $k–i+1$ rounds ($k$ is the number of the items in the sequence). In each round of the policy matching, our protocol considers all credentials an entity has and gives priority to the credential whose access threshold is lower, so we can find the negotiation result sequences for $A$ and $B$ ($<n_{A1}, \ldots, n_{Ai}, \ldots>$ and $<n_{B1}, \ldots, n_{Bi}, \ldots>$) which can satisfy Eq. (4).

$$\begin{cases} \sum_{i=1}^{x} p_{cre_{a_i}} \ge n_{B(n-x+1)} \\ \sum_{i=1}^{x} p_{cre_{b_i}} \ge n_{A(m-x+1)} \end{cases} \tag{4}$$

Therefore, for any $x \in \{1, 2, \ldots, k\}$, our protocol can find a feasible credential set of $A$ ($set_x$) which can satisfy the policy of accessing $cre_{b_x}$ or $resource$. Moreover, we can ensure $n_{B(n-x+1)} \le \sum_{cre \in set_x} p_{cre} \le \sum_{i=1}^{x} p_{cre_{a_i}}$. For the same reason, our protocol can find the feasible credential set of $B$ to satisfy the policy of accessing $A$'s credentials.

In conclusion, our protocol can find a feasible negotiation strategy whenever such a strategy exists.

**Lemma 4:** Assume $A$ and $B$ carry out an interaction. If the access threshold of $B$'s resource($R$) is increased, $A$ will disclose more credentials.

***Proof*:** We denote $T_R$ and $T'_R$ as the threshold of $R$ before and after increase separately. $A$'s responses to $T_R$ and $T'_R$ are denoted as $p$ and $p'$. We use reduction ad absurdum to prove the lemma. Assume if $T_R < T'_R$, there is $p > p'$. According to policy negotiation protocol, there are $\sum_{ci \in cre \cap a_{ci} \le a_{cp}} a_{ci} \ge T_R$ and $\sum_{ci \in cre \cap a_{ci} \le a_{cp}} a_{ci} \ge T'_R \cap \sum_{ci \in cre \cap a_{ci} < a_{cp}} a_{ci} \ge T'_R$ ($cre$ is $A$'s cre-

dential set), so we have $\sum_{ci \in cre \cap a_{ci} \le a_{cp}} a_{ci} \ge T'_R$. If $p > p'$, we have $\sum_{ci \in cre \cap a_{ci} \le a_{cp}} > a_{ci} \sum_{ci \in cre \cap a_{ci} \le a_{cp'}} a_{ci}$, so $\sum_{ci \in cre \cap a_{ci} \le a_{cp'}} a_{ci} < T'_R$. Now, there is a contradiction, so the assumption $p > p'$ is false.

**Theorem 2:** If there exist more than one successful trust negotiation strategies between two participants, our protocol will find the strategy which discloses the minimal privacy information.

***Proof*:** We know $\sum_{i=1}^{n} a_{ci} - L$ ($L$ is the last item in an entity's updated negotiation result sequence) is the capacity of the knapsack in each round of credential determination and exchange. The smaller the $L$, the bigger the capacity of the knapsack, the less the privacy information ($\sum_{i=1}^{n} x_i p_i$ in Eq. (1)) disclosed. It is obviously that in the policy negotiation phase, each item in the negotiation result sequence is the minimal trust score of the credentials an entity need from its counterparty, so our protocol discloses the minimal amount of privacy information. ❑

From the above lemmas and theorems, we know that the proposed protocol is a complete negotiation strategy. In addition, our protocol can find the optimal feasible strategy which ensures the minimal amount of privacy information will be disclosed.

### 5.2 Security

A protocol is defined as secure if it implements a function $f$, such that the information learned by engaging in the protocol can be learned in an ideal implementation where the functionality is provided by a trusted oracle. This definition follows the definition given by Goldreich [31] for private multi-party computation. We define our security model as a semi-honest model. Adversaries are honest but try to compute additional information other than what can be inferred from their input and output. Let $A$ be one of the two participants in our protocol. We use $view_A$ to represent all the information that $A$ has during the protocol. The protocol is secure against a semi-honest $A$, if and only if there exists an algorithm that can simulate $view_A$. We denote $A_I$ and $A_O$ as $A$'s input and output. If there is an algorithm $ALG_A$ such that $view_A$ is indistinguishable from $ALG_A(A_I, A_O)$, the protocol is secure.

**Theorem 3:** The policy matching method (Algorithm 1) is secure in the semi-honest adversarial model.

***Proof*:** We take the sever-side policy matching as an example. We must show that the server's view and the client's view are simulatable from their input and output alone. The server's view is the interaction in the secure two-party maximum protocol and the client's response to the server's policy (fail signal or $i$ mentioned in Algorithm 1). From [18]'s Lemma 6 we know server's output in secure two-party maximum protocol is computationally indistinguishable from the real view. The client's response to the server's policy is the output of the server, so it is simulatable. The client's view includes two things: (1) the client's response to the server's policy which is just the output information and thus is trivially simulatable; (2) the interaction from the secure two-party maximum

protocol, which is simulatable according to the Lemma in [18]. ❑

Authors in [18] proves that the privacy two-party maximum protocol is secure in the semi-honest adversarial. It also proves the security of the Fingerprint protocol for determining the final disclosed credential in the semi-honest adversarial model, so our trust negotiation protocol is secure in the semi-honest adversarial model.

In addition, it is obvious that our protocol will not disclose any unnecessary policies in the negotiation process. The policy negotiation process computes the minimal trust score of the credential set an entity $A$ should disclose in order to access its counterparty $B$'s resource or credential subset, so $A$ can only know its querying resource's access policy or one of $B$'s credential subset's access policy without knowing the access policy of a specific credential in that credential set. In the credential determination phase, the credential selection considers the privacy protection, so the final selected credential set's trust score may bigger than the corresponding item in the negotiation result sequence obtained in policy negotiation phase. As a result, the access policy of the final selected credentials also won't be disclosed.

Because the credential determination will be carried out if the policy negotiation is success, no credential will be disclosed unless there is a feasible negotiation strategy.

In Eager Strategy, since the client and the server negotiate by directly disclosing credentials no matter whether the negotiation will succeed and the credential is necessary, the unnecessary credentials will be disclosed, but there is no unnecessary policies disclosed. Parsimonious Strategy is not complete and has the difficulty of deciding when the negotiation should fail and stop.

### 5.3 Performance

The efficiency of a negotiation protocol includes two aspects: the computational and the communication cost. Below we analyze the performance of the proposed protocol in terms of the two aspects. The cost of the attribute matching phase is a constant, so we only consider the cost in the policy negotiation and credential determination phases. We assume $m_c$ and $m_s$ are the number of credentials of the client and the server.

**Theorem 4:** The worst-case computational complexity of our protocol is $O(2 \times \max\{m_c, m_s\} \times T_{resource} + \max\{m_c, m_s\}^2 \times (\max\{\sum_{i=1}^{m_c} a_{ci}, \sum_{i}^{m_s} a_{ci}\} \times n - T_{resource}))$. $T_{resource}$ is the access threshold of the requesting resource.

***Proof*:** The computational complexity includes two parts: the computational cost of the policy negotiation protocol and the credential determination protocol. In the policy negotiation phase, the client-side and server-side policy matching will be carried out at most $2 \times T_{resource}$ times. The computational cost of policy matching algorithm is $O(\max\{m_c, m_s\})$. In credential determination phase, the cost of fingerprint protocol is $O(\max\{m_c, m_s\}^2) \times T')$ ($T'$ is the marginal threshold in credential determination phase).

Because $\sum_{i=1}^{n} T_i' \le \max\{\sum_{i=1}^{m_c} a_{ci}, \sum_{i}^{m_s} a_{ci}\} \times n - T_{resource}$ ($n$ is the amount of the execution of the Fingerprint protocol and $T_i'$ is the marginal threshold in the $i$th execution of the fingerprint protocol), the computational cost of the credential determination is $O(\max\{m_c,$

$m_s\}^2 \times (\max\{\sum_{i=1}^{m_c} a_{ci}, \sum_{i}^{m_s} a_{ci}\} \times n - T_{resource}))$. Hence, the overall computational cost is $O(2\times\max\{m_c, m_s\} \times T_{resource} + \max\{m_c, m_s\}^2 \times (\max\{\sum_{i=1}^{m_c} a_{ci}, \sum_{i}^{m_s} a_{ci}\} \times n - T_{resource}))$.

**Theorem 5:** The worst-case communication complexity of our protocol is $O(2\times T_{resource} + \max\{m_c, m_s\} \times (T_{resource} + 1) + \sum_{i=1}^{m_c} c_i + \sum_{i=1}^{m_s} s_i)$. $c_i$ and $s_i$ are the number of bits of the client's and the server's $i$th credentials separately.

***Proof*:** In the policy negotiation phase, the policy matching algorithm uses the secure two-party maximum protocol whose communication complexity is a constant. The policy matching algorithm is implemented at most $2\times T_{resource}$ times, so the communication cost is $O(2\times T_{resource})$. In the credential determination and exchange phase, the communication cost of establishing the dynamic programming table is $O(\max\{m_c, m_s\} \times T_{resource})$. Once the dynamic programming table is established, the server only needs to send the cipher text of $M_{n,T'}$ to the user. The size of $M_{n,T'}$ is the size of the sum of privacy scores $\{a_{ci}\}$ and the expanded $n$ additional binary bits. We assume the privacy scores before expansion are bounded by a constant, so the communication cost is $O(n)$, where $n$ is the number of the credentials an entity has, so $n$ won't bigger than $\max\{m_c, m_s\}$. The communication cost of the credential exchange is $O(\sum_{i=1}^{m_c} c_i + \sum_{i=1}^{m_s} s_i)$. Thus, the overall communication cost of our protocol is $O(2\times T_{resource} + \max\{m_c, m_s\} \times (T_{resource} + 1) + \sum_{i=1}^{m_c} c_i + \sum_{i=1}^{m_s} s_i)$.

## 6. SIMULATION RESULTS

For evaluating the efficiency of our protocol, we implemented it using C++. The experiment platform is Windows7 Professional 32bit, Intel(R) Core 2 Duo T5870 2.00 GHz CPU. We assume there are two devices want to interact with each other. They assign the access threshold to each of their credentials and resource. The time for the calculation in policy negotiation and credential determination phase was measured. Since the amount of the credentials and access threshold of the requiring resource affect the computational cost, we used different credential scales and access thresholds to test it.

Fig. 2 shows the computation cost of the policy negotiation protocol. We can see that given the resource access threshold, the computation time increases with the increase of the number of the credentials. However the average computation complexity is smaller than the worst-case computation complexity given in previous section. This is because in most cases, the policy matching process doesn't need to run $2\times T_{resource}$ times.

Fig. 3 shows the computation cost of the credential determination phase. We can see that the computation cost increases with the increase of the number of credentials and the resource's access threshold. The average computation cost is also less than the worst-case complexity. It is obvious that in the whole negotiation process, credential determination consumes most of the computation time because of the encryption and the dynamic programming table establishing. Therefore, our protocol can save the computation time considerably if there is no feasible negotiation strategy between two entities, because they won't carry out the following process. We can see that when the credential

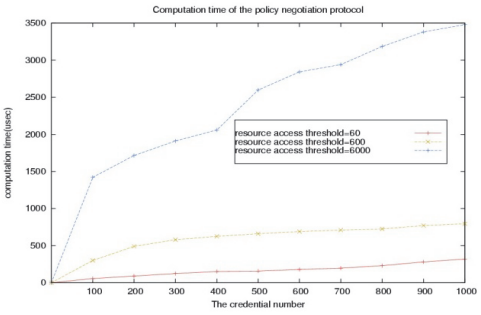



Fig. 2. Computation cost of the policy negotiation protocol.

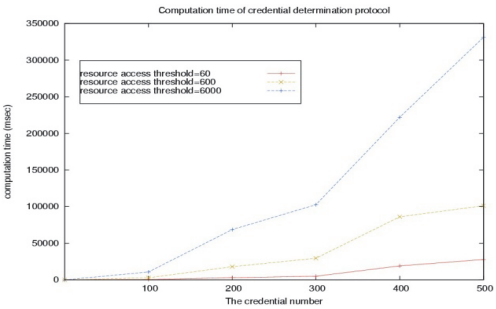



Fig. 3. Computation cost of the credential determination protocol.

scale and the resource's access threshold increase to a certain degree, the computation cost of the credential determination process will become pretty high. Therefore, the protocol is not suitable for an entity with a pretty large amount of credentials.

## 7. CONCLUSIONS

In this paper, we propose a trust negotiation method for devices communication in smart city. We aim to find a feasible trust negotiation strategy which can satisfy both the evolved devices' access policy with minimal amount of privacy disclosed. We consider the credential disclosure problem as a 0/1 knapsack problem which can be solved by dynamic programming. A secure two-party maximum protocol is adopted, so that devices cannot know their counterparties' policies and other privacy information. Theoretical analysis shows our protocol is feasible and complete. In addition, it won't disclose any credentials until two parties can ensure there is a feasible negotiation strategy between them. The experiments illustrate that most of the computation cost of our protocol is consumed in the credential determination phase. When the credential scale increases to a certain degree, the computation cost of the credential determination procedure will become too high. So reducing it is crucial for this approach to be widely used in practical situations. It may be usable in such situations that the credential scale is not so large or that devices negotiate with servers on behalf of their users when they are off-line. In the future, we will improve the protocol to make it suitable for large credential scale.

## REFERENCES

1. J. Zhu and J. Ma, "A new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Consumer Electronics*, Vol. 50, 2004, pp. 231-235.
2. P. Guo, J. Wang, B. Li, and S. Lee, "A variable threshold-value authentication architecture for wireless mesh networks," *Journal of Internet Technology*, Vol. 15, 2014, pp. 929-936.
3. C. D. Jensen, "The importance of trust in computer security," in *Proceedings of IFIP International Conference on Trust Management*, 2014, pp. 1-12.

4. H. B. Zhang, Y. Wang, X. Z. Zhang, and E.-P. Lim, "ReputationPro: the efficient approaches to contextual transaction trust computation in E-commerce environments," *ACM Transactions on the Web*, Vol. 9, 2015, p. 49.
5. T. H. Ma, J. J. Zhou, M. L. Tang, Y. Tian, A. Al-dhelaan, M. Al-rodhaan, and S. Y. Lee, "Social network and tag sources based augmenting collaborative recommender system," *IEICE Transactions on Information and Systems*, Vol. E98-D, 2015, pp. 902-910.
6. L. Xu and Y. Zhang, "A new reputation-based trust management strategy for clustered ad hoc networks," in *Proceedings of IEEE International Conference on Networks Security*, *Wireless Communications and Trusted Computing*, 2009, pp. 116-119.
7. A. Maña, H. Koshutanski, and E. J. Pérez, "A trust negotiation based security framework for service provisioning in load-balancing clusters," *Computers and Security*, Vol. 31, 2012, pp. 4-25.
8. J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in *Proceedings of the 12th ACM Conference on Computer and Communications Security*, 2005, pp. 46-57.
9. W. H. Winsborough, K. E. Seamons, and V. E. Jones, "Automated trust negotiation," in *Proceedings of IEEE DARPA Information Survivability Conference and Exposition*, Vol. 1, 2000, pp. 88-102.
10. J. Lei, B. Zhang, and X. Fang, "Trust vector-based sensitive information protecting scheme in automatic trust negotiation," in *Proceedings of IEEE International Conference on Computer Science and Network Technology*, Vol. 2, 2011, pp. 735-738.
11. P. Bonatti, J. L. De Coi, D. Olmedilla, *et al.* "A rule-based trust negotiation system," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, 2010, pp. 1507-1520.
12. A. Maña, H. Koshutanski, and E. J.Pérez, "A trust negotiation based security framework for service provisioning in load-balancing clusters," *Computers and Security*, Vol. 31, 2012, pp. 4-25.
13. H. Lin, L. Xu, J. Gao, *et al.,* "A subjective logic based dynamic trust mechanism for voice over internet protocol (VOIP) over wireless mesh networks (WMNs)," *Scientific Research and Essays*, Vol. 18, 2011, pp. 3873-3884.
14. N. Li, J. C. Mitchell, and W. H. Winsborough, "Design of a role-based trust-management framework," in *Proceedings of IEEE Symposium on Security and Privacy*, 2002, pp. 114-130.
15. E. Bertino, E. Ferrari, and A. C. Squicciarini, "Privacy-preserving trust negotiations," *Privacy Enhancing Technologies*, 2005, pp. 283-301.
16. J. M. Seigneur and C. D. Jensen, "Trading privacy for trust," *Trust Management*, Springer, Berlin, Heidelberg, 2004, pp. 93-107.
17. Seamons, E. Kent, *et al.*, "Protecting privacy during on-line trust negotiation," *Privacy Enhancing Technologies*, Springer, Berlin, Heidelberg, 2003, pp. 129-143.
18. D. Yao, K. B. Frikken, M. J. Atallah, *et al.*, "Point-based trust: Define how much privacy is worth," *Information and Communications Security*, Springer, Berlin, Heidelberg, 2006, pp. 190-209.
19. Y. Zhang and T. Ishaya, "An XML-based protocol for improving trust negotiation between web services," in *Proceedings of the 27th Annual ACM Symposium on Ap-*

*plied Computing*, 2012, pp. 1947-1954.
20. A. K. Adams, A. J. Lee, and D. Mossé, "Receipt-mode trust negotiation: efficient authorization through outsourced interactions," in *Proceedings of the 6th ACM Symposium on Information*, *Computer and Communications Security*, 2011, pp. 430-434.
21. J. Vaidya, V. Atluri, B. Shafiq, *et al.*, "Privacy-preserving trust verification," in *Proceedings of the 15th ACM Symposium on Access Control Models and Technologies*, 2010, pp. 139-148.
22. Y. Li, N. Cuppens-Boulahia, J. M. Crom, *et al.*, "Reaching agreement in security policy negotiation," in *Proceedings of the 13th IEEE International Conference on Trust*, *Security and Privacy in Computing and Communications*, 2014, pp. 98-105.
23. Y. Zhang and D. Mundy, "Remembrance of local information status for enforcing robustness of policy-exchanged strategies for trust negotiation," in *Proceedings of the 13th IEEE International Conference on Trust*, *Security and Privacy in Computing and Communications*, 2014, pp. 106-113.
24. A. B. C. Douss, S. Ayed, R. Abassi, *et al.* "Trust negotiation based approach to enforce manet routing security," in *Proceedings of the 10th International Conference on Availability, Reliability and Security*, 2015, pp. 360-366.
25. U. Premarathne, I. Khalil, Z. Tari, *et al.*, "Cloud-based utility service framework for trust negotiations using federated identity management," *IEEE Transactions on Cloud Computing*, 2015.
26. A. C. C. Yao, "How to generate and exchange secrets," in *Proceedings of the 27th IEEE Annual Symposium on Foundations of Computer Science*, 1986, pp. 162-167.
27. T. H. Cormen, C. E. Leiserson, R. L. Rivest, *et al.*, *Introduction to Algorithms*, MIT Press, Cambridge, 2001.
28. K. B. Frikken and M. J. Atallah, "Privacy preserving route planning," in *Proceedings of ACM Workshop on Privacy in the Electronic Society*, 2004, pp. 8-15.
29. C. Gentry, "A fully homomorphic encryption scheme," Ph.D. Thesis, Department of Computer Science, Stanford University, 2009.
30. T. Yu, X. Ma, and M. Winslett, "PRUNES: an efficient and complete strategy for automated trust negotiation over the Internet," in *Proceedings of the 7th ACM Conference on Computer and Communications Security*, 2000, pp. 210-219.
31. O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press, UK, 2009.

**Jing-Jing Guo (郭晶晶)** received the M.Sc. and Ph.D. degrees in Computer Science from Xidian University, Xi'an, China, in 2012 and 2015, respectively. She is currently an Assistant Professor in the School of Cyber Engineering at Xidian University. Her research interests include trust management, social networks, access control and information security.

**Jian-Feng Ma (马建峰)** received the ME and Ph.D. degrees in Computer Software and Communications Engineering from Xidian University, in 1988 and 1995, respectively. He is now a Full Professor and Ph.D. supervisor in Xidian University and a member of China Computer Federation. His main research interests include information security, coding theory, and cryptography. He is a member of the IEEE.

**Xing-Hua Li (李兴华)** received the ME and Ph.D. degrees in Computer Science from Xidian University, in 2004 and 2007, respectively. He is now a Full Professor and Ph.D. supervisor in Xidian University. His main research interests include wireless networks security, privacy protection, cloud computing, software defined network, and security protocol formal methodology. He is a member of the IEEE.

**Jun-Wei Zhang (张俊伟)** received the Ph.D. degree in Computer Architecture from Xidian University. He is now an Associate Professor of Xidian University. His research interests include cryptography and information security.

**Tao Zhang (张涛)** received M.Sc. and Ph.D. degrees in Computer Science from Xidian University, Xi'an, China, in 2011 and 2015, respectively. He is currently an Assistant Professor in the School of Computer Science at Xidian University. His research interests include trust management, social networks, web services and information security.